MULTIMEDIA UNIVERSITY

STUDENT ID NO

MULTIMEDIA UNIVERSITY

FINAL EXAMINATION

TRIMESTER 2, 2018/2019

TSN3151 – PARALLEL PROCESSING

(All Sections / Groups)

13 MARCH 2019
2:30 p.m. – 4:30 p.m.
(2 Hours)

INSTRUCTIONS TO STUDENTS

1. This question paper consists of 7 pages only (including this page).
2. **This is an open-book exam**, so the questions are more in-depth and there are fewer questions than normal exams. Attempt **ALL** questions. The distribution of the marks for each question is given.
3. Print all your answers in the Answer Booklet provided.
4. State all assumptions clearly.

Question 1 (10 Marks)

a) Using the e-cube routing (also known as a deadlock free routing algorithm), determine the route taken in a 6-dimensional Hypercube network from Node 8 to Node 53. Your final answer should be stated in Node (decimal) number, show your steps clearly.

[5 marks]

b) Describe what is a bisection bandwidth in terms of computer networking and what is the bisection bandwidth of an 8x6 mesh?

[1 mark]

c) In computer network embedding, what is meaning of dilation? Determine the dilation of embedding a torus to a mesh network, provide an explanation for your answer.

[2 marks]

d) Discuss whether the e-cube routing (as in question (a) above) can be used for routing between nodes in a torus network?

[2 marks]

Continued ...

Question 2 (10 Marks)

Study the code below.

```
 1 #include <stdio.h>
 2 #include <stdlib.h>
 3 #include <string.h>
 4 #include "mpi.h"
 5
 6 int main(int argc, char**argv) {
 7   int r, np;
 8   int *a, i;
 9
10   MPI_Init(&argc, &argv);
11   MPI_Comm_size(MPI_COMM_WORLD, &np);
12   MPI_Comm_rank(MPI_COMM_WORLD, &r);
13
14   if ( np < 3 ){
15     printf("Sorry, this only works if we have at least 3 processes.\n");
16     // missing code
17     exit(1);
18   }
19
20   // Allocate an array of np integers
21   a = (int *) malloc(sizeof(int)*np);
22
23   for (i=0; i<np; i++)
24     a[i] = i+(r*10);
25
26   MPI_Bcast(&(a[np-1]), 1, MPI_INT, np-1, MPI_COMM_WORLD );
27   MPI_Allreduce(&(a[1]), &(a[2]),1,MPI_INT,MPI_SUM,MPI_COMM_WORLD);
28
29   MPI_Finalize();
30 }
```

a) Explain why lines 14-18 are necessary and what is the missing code in line 16?

[2 marks]

b) If np = 6, what does the array "a" contain for process 0 (rank 0) after line 24 is executed? Give your answer using the following format;

Index	0	1	2	3	4	5
Value						

[2 marks]

c) Explain what line 26 does. You may explain in terms of np=6.

[1 mark]

d) Explain what happens when line 27 is executed. You may explain in terms of np=6.

[1 mark]

Continued ...

e) If np=8, what do all the arrays in all the processes contain after line 27? Give your answer using the following format;

	Array "a" Index						
Process	0	1	2	3	4	5	6
0							
1							
2							
3							
4							
5							
6							

[4 marks]

Continued ...

Question 3 (10 Marks)

Study the serial code below. It is a two dimension function, `f(x,y)`, that calculates the volume below the surface created by the function `f()` at x between 1.75 and 2.15 and y between 4.2 and 5.4 using 1,000 slices for both x and y using rectangular quadrature.

```
#include <stdio.h>
double f(double x, double y);        // The function to integrate
#define Ax 1.75                      // Start point for x
#define Bx 2.15                      // End point for x
#define Ay 4.2                       // Start point for y
#define By 5.4                       // End point for y
#define SLICES 1000

int main(int argc, char *argv[]) {
  double xregion = Bx-Ax;            // Region for x
  double yregion = By-Ay;            // Region for y;
  double xd = xregion/SLICES;        // Each small slice for x
  double yd = yregion/SLICES;        // Each small slice for y
  double total = 0;
  double x,y;

  for (y=Ay+yd/2; y<By; y+=yd)
    for (x=Ax+xd/2; x<Bx; x+=xd)
      total += f(x,y);

  total = total * xd * yd;
  printf("The definite integral is %f.\n", total);
}
```

a) Convert the serial program provided above into a pthread program so that 4 threads do 250 slices each.

[7 marks]

b) What is this type of parallelism called?

[1 mark]

c) What is the expected speed up expected from the program once it is parallelize and executed on *p* processors? State your assumptions in your discussion.

[2 marks]

Continued ...

Question 4 (10 Marks)

Study the serial code below which computes

$$\int_{x=0}^{1} \sin(x) + x^2 \, dx$$

using 1,000,000 intervals.

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

int main(int argc, char*argv[]) {
  int numPartitions;
  double step = 1.0/1000000;        // Using 1,000,000 partitions
  double x, total=0;
  int i;

  for (i=0; i<1000000; i++) {
    x=(step/2) + step*i;
    total = total + sin(x)+x*x;
  }
  printf("The result is %f\n",total * step);
}
```

a) Convert the code to use the Message Passing Interface (MPI) library.

[6 marks]

Continued ...

b) Study the code below which computes the value of π. Explain why the answer will be different each time the program is executed and correct the code appropriately.

```
#include <stdio.h>
#include <math.h>
#include <pthread.h>
#define NUM_RECT 10000000
#define NUMTHREADS 4
double gPi = 0.0, x;     //  global accumulator for areas computed

void *Area(void *pArg){
  int myNum = *((int *)pArg);
  double h = 2.0 / NUM_RECT;

  for (int i = myNum; i < NUM_RECT; i += NUMTHREADS){
    x = -1 + (i + 0.5f) * h;
    gPi += sqrt(1.0 - x*x) * h;
  }
  return 0;
}

int main(int argc, char **argv) {
  pthread_t tHandles[NUMTHREADS];
  int tNum[NUMTHREADS], i;

  for ( i = 0; i < NUMTHREADS; ++i ) {
    tNum[i] = i;
    pthread_create(&tHandles[i], NULL, Area, (void*)&tNum[i]);
  }

  for ( i = 0; i < NUMTHREADS; ++i ) {
    pthread_join(tHandles[i], NULL);
  }
  gPi *= 2.0;

  printf("Computed value of Pi:  %12.9f\n", gPi );

  return 0;
}
```

[4 marks]

End of Page